(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 055 990 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.11.2000 Bulletin 2000/48

(51) Int Cl.[7]: **G06F 1/00**

(21) Application number: **99304165.6**

(22) Date of filing: **28.05.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant: **Hewlett-Packard Company**
Palo Alto, California 94304-1112 (US)

(72) Inventors:
- **Proudler, Graeme**
  Bristol BS34 8XQ (GB)
- **Balacheff, Boris**
  Bristol BS31 2HJ (GB)
- **Pearson, Siani**
  Bristol BS9 3PZ (GB)
- **Chan, David**
  California CA 95030 (US)

(74) Representative: **Lawman, Matthew John Mitchell**
Hewlett-Packard Limited,
IP Section,
Building 3,
Filton Road
Stoke Gifford, Bristol BS34 8QZ (GB)

(54) Event logging in a computing platform

(57) There is disclosed a computer entity having a trusted component which compiles an event log for events occurring on a computer platform. The event log contains event data of types which are pre-specified by a user by inputting details through a dialogue display generated by the trusted component. Items which can be monitored include data files, applications drivers and the like. The trusted component operates through a monitoring agent which may be launched onto the computer platform. The monitoring agent may be periodically interrogated to make sure that it is operating correctly and responding to interrogations by the trusted component.

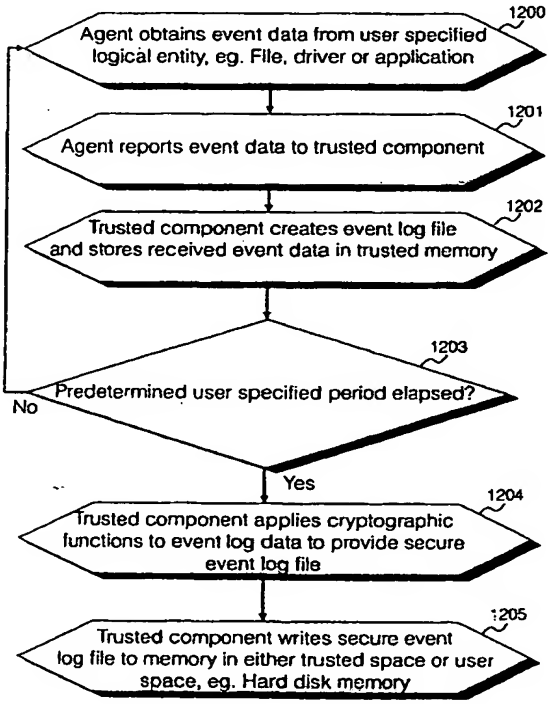

Fig. 12

EP 1 055 990 A1

Description

Field of the Invention

[0001] The present invention relates to security monitoring of computer platforms, and particularly, although not exclusively, to monitoring of events and operations occurring on data files, applications, drivers and like entities on a computer platform.

Background to the Invention

[0002] Conventional prior art mass market computing platforms include the well-known personal computer (PC) and competing products such as the Apple Macintosh™, and a proliferation of known palm-top and laptop personal computers. Generally, markets for such machines fall into two categories, these being domestic or consumer, and corporate. A general requirement for a computing platform for domestic or consumer use is a relatively high processing power, Internet access features, and multi-media features for handling computer games. For this type of computing platform, the Microsoft Windows® '95 and '98 operating system products and Intel processors dominate the market.

[0003] On the other hand, for business use, there are a plethora of available proprietary computer platform solutions available aimed at organizations ranging from small businesses to multi-national organizations. In many of these applications, a server platform provides centralized data storage, and application functionality for a plurality of client stations. For business use, other key criteria are reliability, networking features, and security features. For such platforms, the Microsoft Windows NT 4.0™ operating system is common, as well as the Unix™ operating system.

[0004] With the increase in commercial activity transacted over the Internet, known as "e-commerce", there has been much interest in the prior art on enabling data transactions between computing platforms, over the Internet. However, because of the potential for fraud and manipulation of electronic data, in such proposals, fully automated transactions with distant unknown parties on a wide-spread scale as required for a fully transparent and efficient market place have so far been held back. The fundamental issue is one of trust between interacting computer platforms for the making of such transactions.

[0005] There have been several prior art schemes which are aimed at increasing the security and trustworthiness of computer platforms. Predominantly, these rely upon adding in security features at the application level, that is to say the security features are not inherently imbedded in the kernel of operating systems, and are not built in to the fundamental hardware components of the computing platform. Portable computer devices have already appeared on the market which include a smart card, which contains data specific to a user, which is input into a smart card reader on the computer. Presently, such smart cards are at the level of being add-on extras to conventional personal computers, and in some cases are integrated into a casing of a known computer. Although these prior art schemes go some way to improving the security of computer platforms, the levels of security and trustworthiness gained by prior art schemes may be considered insufficient to enable widespread application of automated transactions between computer platforms. Before businesses expose significant value transactions to electronic commerce on a widespread scale, they may require greater confidence in the trustworthiness of the underlying technology.

[0006] In the applicant's co-pending disclosures 'Trusted Computing Platform', filed at the European Patent Office on 15 February 1999, the entire contents of which are incorporated herein by reference, and 'Computing Apparatus and Methods of Operating Computing Apparatus', there is disclosed a concept of a 'trusted computing platform' comprising a computing platform which has a 'trusted component' in the form of a built-in hardware and software component. Two computing entities each provisioned with such a trusted component, may interact with each other with a high degree of 'trust'. That is to say, where the first and second computing entities interact with each other the security of the interaction is enhanced compared to the case where no trusted component is present, because:

- A user of a computing entity has higher confidence in the integrity and security of his/her own computer entity and in the integrity and security of the computer entity belonging to the other computing entity.
- Each entity is confident that the other entity is in fact the entity which it purports to be.
- Where one or both of the entities represent a party to a transaction, e.g. a data transfer transaction, because of the in-built trusted component, third party entities interacting with the entity have a high degree of confidence that the entity does in fact represent such a party.
- The trusted component increases the inherent security of the entity itself, through verification and monitoring processes implemented by the trusted component.
- The computer entity is more likely to behave in the way it is expected to behave.

[0007] Prior art computing platforms have several problems which need to be overcome in order to realize the potential of the applicants' above disclosed trusted component concept. In particular,

- The operating status of a computer system or plat-

form and the status of the data within the platform or system is dynamic and difficult to predict. It is difficult to determine whether a computer platform is operating correctly because the state of the computer platform and data on the platform is constantly changing and the computer platform itself may be dynamically changing.

- From a security point of view, commercial computer platforms, in particular client platforms, are often deployed in environments which are vulnerable to unauthorized modification. The main areas of vulnerability include modification by software loaded by a user, or by software loaded via a network connection. Particularly, but not exclusively, conventional computer platforms may be vulnerable to attack by virus programs, with varying degrees of hostility.

- Computer platforms may be upgraded or their capabilities extended or restricted by physical modification, i.e. addition or deletion of components such as hard disk drives, peripheral drivers and the like.

[0008] It is known to provide certain security features in computer systems, embedded in operating software. These security features are primarily aimed at providing division of information within a community of users of the system.

[0009] In the known Microsoft Windows NT™ 4.0 operating system, there also exists a monitoring facility called "system log event viewer" in which a log of events occurring within the platform is recorded into an event log data file which can be inspected by a system administrator using the windows NT operating system software. This facility goes someway to enabling a system administrator to security monitor pre-selected events. The event logging function in the Windows NT™ 4.0 operating system is an example of system monitoring.

[0010] However, in terms of overall security of a computer platform, a purely software based system is vulnerable to attack, for example by viruses. The Microsoft Windows NT™ 4.0 software includes a virus guard software, which is preset to look for known viruses. However, virus strains are developing continuously, and the virus guard software will not guard against unknown viruses.

[0011] Further, prior art monitoring systems for computer entities focus on network monitoring functions, where an administrator uses network management software to monitor performance of a plurality of network computers. Also, trust in the system does not reside at the level of individual trust of each hardware unit of computer platform in a system.

Summary of the Invention

[0012] Specific implementations of the present invention provide a computer platform having a trusted component which is physically and logically distinct from a computer platform. The trusted component has the properties of unforgability, and autonomy from the computer platform with which it is associated. The trusted component monitors the computer platform and thereby may provide a computer platform which is monitored on an individual basis at a level beneath a network monitoring or system monitoring level. Where a plurality of computer platforms are networked or included in the system, each computer platform may be provided with a separate corresponding respective trusted component.

[0013] Specific implementations of the present invention may provide a secure method of monitoring events occurring on a computer platform, in a manner which is incorruptible by alien agents present on the computer platform, or by users of the computer platform, in a manner such that if any corruption of the event log takes place, this is immediately apparent.

[0014] . According to a first aspect of the present invention there is provided a computer entity comprising a computer platform comprising a data processor and at least one memory device; and a trusted component, said trusted component comprising a data processor and at least one memory device; wherein said data processor and said memory of said trusted component are physically and logically distinct from said data processor and memory of said computer platform; and means for monitoring a plurality of events occurring on said computer platform.

[0015] Preferably said monitoring means comprises a software agent operating on said computer platform, for monitoring at least one event occurring on said computer platform, and reporting said event to said trusted component.

[0016] Said software agent may comprise a set of program code normally resident in said memory device of said trusted component, said code being transferred into said computer platform for performing monitoring functions on said computer platform.

[0017] Preferably said trusted component comprises an event logging component for receiving data describing a plurality of events occurring on said computer platform, and compiling said event data into a secure event data.

[0018] Preferably said event logging component comprises means for applying a chaining function to said event data to produce said secure event data.

[0019] Selections of events and entities to be monitored may be selected by a user by operating a display interface for generating an interactive display comprising: means for selecting an entity of said computer platform to be monitored; and means for selecting at least one event to be monitored.

[0020] The monitoring means may further comprise prediction means for predicting a future value of at least one selected parameter.

[0021] Preferably the computer entity further comprises a confirmation key means connected to said trusted component, and independent of said computer platform, for confirming to said trusted component an authorisation signal of a user.

[0022] Entities to be monitored may include a data file; an application; or a driver component.

[0023] According to a second aspect of the present invention there is provided a computer entity comprising a computer platform having a first data processor and a first memory device; and a trusted monitoring component comprising a second data processor and a second memory device, wherein said trusted monitoring component stores an agent program resident in said second memory area, wherein said agent program is copied to said first memory area for performing functions on behalf of said trusted component, under control of said first data processor.

[0024] According to a third aspect of the present invention there is provided a computer entity comprising a computer platform comprising a first data processor and a first memory device; a trusted monitoring component comprising a second data processor and a second memory device; a first computer program resident in said first memory area and operating said first data processor, said first computer program reporting back events concerning operation of said computer platform to said trusted monitoring component; and a second computer program resident in said second memory area of said trusted component, said second program operating to monitor an integrity of said first program.

[0025] Said computer program may monitor an integrity of said first computer program by sending to said first computer program a plurality of interrogation messages, and monitoring a reply to said interrogation messages made by said first computer program.

[0026] Preferably said interrogation message is sent in a first format, and returned in a second format, wherein said second format is a secure format.

[0027] According to a fourth aspect of the present invention there is provided a method of monitoring a computer platform comprising a first data processor and a first memory means, said method comprising the steps of reading event data describing events occurring on at least one logical or physical entity comprising said computer platform; securing said event data in a second data processing means having an associated second memory area, said second data processing means and said second memory area being physically and logically distinct from said first data processing means and said first memory area, such that said secured event data cannot be altered without such alteration being apparent.

[0028] A said event to be monitored may be selected from the set of events: copying of a data file; saving a data file; renaming a data file; opening a data file; overwriting a data file; modifying a data file; printing a data file; activating a driver device; reconfiguring a driver device; writing to a hard disk drive; reading a hard disk drive; opening an application; closing an application.

[0029] A said entity to be monitored may be selected from the set: at least one data file stored on said computer platform; a driver device of said computer platform; an application program resident on said computer platform.

[0030] The entity may be continuously monitored over a pre-selected time period, or the entity may be monitored until such time as a pre-selected event occurs on the entity. The entity may be monitored for a selected event until a pre-determined time period has elapsed.

[0031] The invention includes a method of monitoring a computer platform comprising a first data processing means and a first memory means, said method comprising the steps of generating an interactive display for selecting at least one entity comprising said computer platform; generating a display of events which can be monitored; generating a display of entities of said computer platform; selecting at least one said entity; selecting at least one said event; and monitoring a said entity for a said event.

[0032] The invention includes a method of monitoring a computer platform comprising a first data processing means and first memory means, said method comprising the steps of storing a monitoring program in a second memory area, said second memory area being physically and logically distinct from said first memory area; transferring said monitoring program from said second memory area to said first memory area; monitoring at least one entity of said computer platform from within said computer platform; and reporting an event data from said monitoring program to said second data processor.

[0033] The invention includes a method of monitoring a computer platform comprising a first data processing and a first memory means, said method comprising the steps of monitoring at least one entity comprising said computer platform from within said computer platform; generating an event data describing a plurality of events occurring on said computer platform; reporting said event data to a second data processing means having an associated second memory means; and processing said event data into a secure format.

Brief Description of the Drawings

[0034] For a better understanding of the invention and to show how the same may be carried into effect, there will now be described by way of example only, specific embodiments, methods and processes according to the present invention with reference to the accompanying drawings in which:

Fig. 1 illustrates schematically a computer entity according to first specific embodiment of the present invention;

Fig. 2 illustrates schematically connectivity of selected components of the computer entity of Fig. 1;

Fig. 3 illustrates schematically a hardware architecture of components of the computer entity of Fig. 1;

Fig. 4 illustrates schematically an architecture of a trusted component comprising the computer entity of Fig. 1;

Fig. 5 illustrates schematically a logical architecture of the computer entity, divided into a monitored user space, resident on the computer platform and a trusted space resident on the trusted component;

Fig. 6 illustrates schematically components of a monitoring agent which monitors events occurring on the computer platform and reports back to the trusted component;

Fig. 7 illustrates schematically logical components of the trusted component itself;

Fig. 8 illustrates schematically process steps carried out for establishing a secure communication between the user and the trusted component by way of a display on a monitor device;

Fig. 9 illustrates schematically process steps for selecting security monitoring functions using a display monitor;

Fig. 10 illustrates schematically a first dialogue box display generated by the trusted component;

Fig. 11 illustrates schematically a second dialogue box display used for entering data by a user;

Fig. 12 illustrates schematically operations carried out by the monitoring agent and the trusted component for monitoring logical and/or physical entities such as files, applications or drivers on the computer platform;

Fig. 13 illustrates schematically process steps operated by the agent and trusted component for continuous monitoring of specified events on the computer platform; and

Fig. 14 illustrates schematically process steps carried out by and interaction between the monitoring agent and the trusted component for implementing the agent on the computer platform, and monitoring the existence and integrity of the agent on the computer platform.

Detailed Description of the Best Mode for Carrying Out the Invention

[0035] There will now be described by way of example the best mode contemplated by the inventors for carrying out the invention. In the following description numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent however, to one skilled in the art, that the present invention may be practiced without limitation to these specific details. In other instances, well known methods and structures have not been described in detail so as not to unnecessarily obscure the present invention.

[0036] In this specification, the term "trusted" when used in relation to a physical or logical component, is used to mean a physical or logical component with which the behavior of that component is predictable and known. Trusted components have a high degree of resistance to unauthorised modification.

[0037] In this specification, the term "computer platform" is used to refer to at least one data processor and at least one data storage means, usually but not essentially with associated communications facilities eg a plurality of drivers, associated applications and data files, and which may be capable of interacting with external entities eg. a user or another computer entity, for example by means of connection to the internet, connection to an external network, or by having an input port capable of receiving data stored on a data storage medium, eg a CD ROM, floppy disk, ribbon tape or the like. The term "computer platform" encompasses the main data processing and storage facility of a computer entity.

[0038] Referring to Fig. 1 herein, there is illustrated schematically one example of a computer entity as previously described in the applicant's European patent application entitled "Trusted Computing Platform", filed 15 February 1999 at the European Patent Office a copy of which is filed herewith, and the entire contents of which are incorporated herein by reference. Referring to Fig. 2 of the accompanying drawings, there is illustrated schematically physical connectivity of some of the components of the trusted computer entity of Fig. 1. Referring to Fig. 3 herein, there is illustrated schematically an architecture of the trusted computer entity of Figs. 1 and 2, showing physical connectivity of components of the entity.

[0039] In general, in the best mode described herein, a trusted computer entity comprises a computer platform consisting of a first data processor, and a first memory means, together with a trusted component which verifies the integrity and correct functioning of the computing platform. The trusted component comprises a second data processor and a second memory means, which are physically and logically distinct from the first data processor and first memory means.

[0040] In the example shown in Figs. 1 to 3 herein, the trusted computer entity is shown in the form of a per-

sonal computer suitable for domestic use or business use. However, it will be understood by those skilled in the art that that this is just one specific embodiment of the invention, and other embodiments of the invention may take the form of a palmtop computer, a laptop computer, a server-type computer, a mobile phone-type computer, or the like and the invention is limited only by the scope of the claims herein. In the best mode example described herein, the computer entity comprises a display monitor 100; a keyboard data entry means 101; a casing 102 comprising a motherboard on which is mounted a data processor; one or more data storage means e.g. hard disk drives; a dynamic random access memory; various input and output ports (not illustrated in Fig. 1); a smart card reader 103 for accepting a user's smart card; a confirmation key 104, which a user can activate when confirming a transaction via the trusted computer entity; and a pointing device, e.g. a mouse or trackball device 105. The trusted computer entity has a trusted component as described in the applicant's previous disclosure and as further described herein.

[0041] Referring to Fig. 2 herein, there are illustrated some of the components comprising the trusted computer entity, including keyboard 100, which incorporates confirmation key 104 and smart card reader 103; a main motherboard 200 on which is mounted first data processor 201 and trusted component 202, an example of a hard disc drive 203, and monitor 100. Additional components of the trusted computer entity, include an internal frame to the casing 102, housing one or more local area network (LAN) ports, one or more modem ports, one or more power supplies, cooling fans and the like (not shown in Fig. 2).

[0042] In the best mode herein, as illustrated in Fig. 3 herein, main motherboard 200 is manufactured comprising a processor 201; and preferably a permanently fixed trusted component 202; a local memory device 300 to the processor, the local memory device being a fast access memory area, e.g. a random access memory; a BIOS memory area 301; smart card interface 305; a plurality of control lines 302; a plurality of address lines 303; a confirmation key interface 306; and a data bus 304 connecting the processor 201, trusted component 202, memory area 300, BIOS memory area 301 and smart card interface 305. A hardware random number generator RNG 309 is also able to communicate with the processor 201 using the bus 304.

[0043] External to the motherboard and connected thereto by data bus 304 are provided the one or more hard disk drive memory devices 203, keyboard data entry device 101, pointing device 105, e.g. a mouse, trackball device or the like; monitor device 100; smart card reader device 103 for accepting a smart card device as described previously; the disk drive(s), keyboard, monitor, and pointing device being able to communicate with processor 201 via said data bus 304; and one or more peripheral devices 307, 308, for example a modem, printer scanner or other known peripheral device.

[0044] Smart card reader 103 is wired directly to smart card interface 305 on the motherboard and does not connect directly to data bus 304. Alternatively, smart card reader 103 may be connected directly to data bus 304. To provide enhanced security confirmation key switch 104 is hard wired directly to confirmation key interface 306 on motherboard 200, which provides a direct signal input to trusted component 202 when confirmation key 104 is activated by a user such that a user activating the confirmation key sends a signal directly to the trusted component, by-passing the first data processor and first memory means of the computer platform.

[0045] Trusted component 202 is positioned logically and physically between monitor 100 and processor 201 of the computing platform, so that trusted component 202 has direct control over the views displayed on monitor 100 which cannot be interfered with by processor 201.

[0046] In one embodiment the confirmation key may comprise a simple switch. Confirmation key 104, and confirmation key driver 306 provide a protected communication path (PCP) between a user and the trusted component, which cannot be interfered with by processor 201, which by-passes data bus 304 and which is physically and logically unconnected to memory area 300 or hard disk drive memory device(s) 203.

[0047] The trusted component lends its identity and trusted processes to the computer platform and the trusted component has those properties by virtue of its tamper-resistance, resistance to forgery, and resistance to counterfeiting. Only selected entities with appropriate authentication mechanisms are able to influence the processes running inside the trusted component. Neither a user of the trusted computer entity, nor anyone or any entity connected via a network to the computer entity may access or interfere with the processes running inside the trusted component. The trusted component has the property of being "inviolate".

[0048] The smart card may comprise a "cash card" or a "crypto card" the functions of which are described in the applicant's above-mentioned previous disclosure "Computing Apparatus and Methods of Operating Computing Apparatus", a copy of which is filed herewith, and the entire content of which is incorporated herein by reference.

[0049] On each individual smart card may be stored a corresponding respective image data which is different for each smart card. For user interactions with the trusted component, e.g. for a dialogue box monitor display generated by the trusted component, the trusted component takes the image data 1001 from the user's smart card, and uses this as a background to the dialogue box displayed on the monitor 100. Thus, the user has confidence that the dialogue box displayed on the monitor 100 is generated by the trusted component. The image data is preferably easily recognizable by a human being in a manner such that any forgeries would be immediately apparent visually to a user. For example, the

image data may comprise a photograph of a user. The image data on the smart card may be unique to a person using the smart card.

[0050] In the best mode herein, a user may specify a selected logical or physical entity on the computer platform, for example a file, application, driver, port, interface or the like for monitoring of events which occur on that entity. Two types of monitoring may be provided, firstly continuous monitoring over a predetermined period, which is set by a user through the trusted component, and secondly, monitoring for specific events which occur on an entity. In particular, a user may specify a particular file of high value, or of restricted information content and apply monitoring of that specified file so that any interactions involving that file, whether authorized or not, are automatically logged and stored in a manner in which the events occurring on the file cannot be deleted, erased or corrupted, without this being immediately apparent.

[0051] Referring to Fig. 4 herein, there is illustrated schematically an internal architecture of trusted component 202. The trusted component comprises a processor 400, a volatile memory area 401; a non-volatile memory area 402; a memory area storing native code 403; and a memory area storing one or a plurality of cryptographic functions, 404, the non-volatile memory 401, native code memory 403 and cryptographic memory 404 collectively comprising the second memory means hereinbefore referred to.

[0052] Trusted component 202 comprises a physically and logically independent computing entity from the computer platform. In the best mode herein, the trusted component shares a motherboard with the computer platform so that the trusted component is physically linked to the computer platform. In the best mode, the trusted component is physically distinct from the computer platform, that is to say it does not exist solely as a sub-functionality of the data processor and memory means comprising the computer platform, but exists separately as a separate physical data processor 400 and separate physical memory area 401, 402, 403, 404. By providing a physically present trusted component, the trusted component becomes harder to mimic or forge through software introduced onto the computer platform. Programs within the trusted component are pre-loaded at manufacture of the trusted component, and are not user configurable. The physicality of the trusted component, and the fact that the user component is not configurable by the user enables the user to have confidence in the inherent integrity of the trusted component, and therefore a high degree of "trust" in the operation and presence of the trusted component on the computer platform.

[0053] Referring to Fig. 5 herein, there is illustrated schematically a logical architecture of the computer entity 500. The logical architecture has a same basic division between the computer platform, and the trusted component, as is present with the physical architecture described in Figs. 1 to 3 herein. That is to say, the trusted component is logically distinct from the computer platform to which it is physically related. The computer entity comprises a user space 504 being a logical space which is physically resident on the computer platform (the first processor and first data storage means) and a trusted component space 513 being a logical space which is physically resident on the trusted component 202. In the user space 504 are one or a plurality of drivers 506, one or a plurality of applications programs 507, a file storage area 508; smart card reader 103; smart card interface 305; and a software agent 511 which operates to perform operations in the user space and report back to trusted component 202. The trusted component space is a logical area based upon and physically resident in the trusted component, supported by the second data processor and second memory area of the trusted component. Confirmation key device 104 inputs directly to the trusted component space 513, and monitor 100 receives images directly from the trusted component space 513. External to the computer entity are external communications networks eg the Internet 501, and various local area networks, wide area networks 502 which are connected to the user space via the drivers 506 which may include one or more modem ports. External user smart card 503 inputs into smart card rear 103 in the user space.

[0054] In the trusted component space, are resident the trusted component itself, displays generated by the trusted component on monitor 100; and confirmation key 104, inputting a confirmation signal via confirmation key interface 306.

[0055] Referring to Fig. 6 herein, within agent 511, there is provided a communications component 601 for communicating with the trusted component 202; and a file monitoring component 600 the purpose of which is to monitor events occurring on specified logical or physical entities, eg data files, applications or drivers on the computer platform, within the user space.

[0056] Referring to Fig. 7 herein, there is illustrated schematically internal components on the trusted component 202 resident in trusted space 513. The trusted component comprises a communications component 700 for communicating with software agent 511 in user space; a display interface component 701 which includes a display generator for generating a plurality of interface displays which are displayed on monitor 100 and interface code enabling a user of the computing entity to interact with trusted component 202; an event logger program 702 for selecting an individual file, application, driver or the like on the computer platform, and monitor the file, application or driver and compile a log of events which occur on the file, application or driver; a plurality of cryptographic functions 703 which are used to cryptographically link the event log produced by event logger component 702 in a manner from which it is immediately apparent if the event log has been tampered with after leaving event logger 702; a set of prediction

algorithms 704 for producing prediction data predicting the operation and performance of various parameters which may be selected by a user for monitoring by the trusted component; and an alarm generation component 705 for generating an alarm when monitored event parameters fall outside pre-determined ranges set by a user, or fall outside ranges predicted by prediction algorithms 704.

[0057] Operation of the computer entity, and in particular operation of trusted component 202 and its interactivity with agent 511 for monitoring of events on the computer platform will now be described.

[0058] Referring to Fig. 8 herein, there is illustrated schematically a set of process steps carried out by the computer entity for generating a dialogue display on monitor 100 and for establishing to a user of the monitor that the trusted component within the computer entity is present and functioning. Firstly, in step 800, a user of the computer entity enters his or her smart card 503 into smart card reader port 103. A pre-stored algorithm on the smart card generates a nonce R1, and downloads the nonce R1 to the trusted component through the smart card reader 103, smart card interface 305 and via data bus 304 to the trusted component 202. The nonce R1 typically comprises a random burst of bits generated by the smart card 503. Smart card 503 stores the nonce R1 temporarily on an internal memory of the smart card in order to compare the stored nonce R1 with a response message to be received from the trusted component. In step 802, the trusted component receives the nonce R1, generates a second nonce R2, concatenates R1 with R2, and proceeds to sign the concatenation R1||R2 using cryptographic functions 703. The process of applying a digital signature in order to authenticate digital data is well known in the art and is described in "Handbook of Applied Cryptography", Menezes Vanoorschot, Vanstone, in sections 1.6 and 1.83. Additionally, an introduction to the use of digital signatures can be found in "Applied Cryptography - Second edition", Schneier, in section 2.6. Trusted component 202 then resends the signed nonces back to the smart card in step 803. The smart card checks the signature on the received message returned from the trusted component in step 804 and compares the nonce contained in the received message with the originally sent nonce R1, a copy of which has been stored in its internal memory. If the nonce returned from the trusted component is different to that from the stored nonce then in step 805 the smart card stops operation in step 806. Difference in nonce's indicates that the trusted component is either not working properly, or there has been some tampering with the nonce data between the smart card reader 103 and trusted component 202 resulting in changes to the nonce data. At this point, smart card 503 does not "trust" the computer entity as a whole because its generated nonce has not been correctly returned by the computer entity.

[0059] If the nonce returned from the trusted component is identical to that as originally sent by the smart card and the comparison of the two R1 nonce's in 805 is successful, in step 807, the smart card then proceeds to retrieve a stored image data from its internal memory, append the nonce R2, sign the concatenation, encrypt the stored image data and send the encrypted image data and the signature to the trusted component via smart card reader 103. The trusted component receives the encrypted image and signature data via smart card reader interface 305, and data bus 304 and in step 808 decrypts the image data and verifies the signature using its cryptographic functions 703, and verifies the nonce R2. The image data is stored internally in the memory area of the trusted component. The trusted component then uses the image data as a background for any visual displays it generates on monitor 100 created by trusted component 203 for interaction with the human user in step 809.

[0060] Referring to Figs. 9 to 11 herein, there will now be described a set of process steps carried out by the computer entity for selecting items to be monitored on the computer platform, and for activating a monitoring session. In step 900, a user selects the security monitoring function by clicking pointing device 105 on an icon presented on a normal operating system view on monitor 100. The icon is generated by a display generator component of display interface 701 of the trusted component 202. Clicking the icon causes the trusted component to generate a dialogue box display on the monitor 100, for example as illustrated in Fig. 10 herein. The dialogue box display on monitor 100 is generated directly by display interface component 701 in a secure memory area of trusted component 202. Display of the image 1001 downloaded from the user's smart card 503 gives a visual confirmation to a user that the dialogue box is generated by the trusted component, since the trusted component is the only element of the computer entity which has access to the image data stored on the smart card. On the security monitoring dialogue box, there is an icon for "file" 1002 which is activated in a file monitoring mode of operation (not described herein) of the computer entity, and an "event" icon 1003 for event monitoring operation. A user selects an event monitoring menu 1100 by clicking the "event" icon 1003 by operating the pointing device 105 on the event icon 1003, in step 902. On activation of the "event" icon, the trusted component generates a second dialogue box comprising an event monitoring menu 1100 which also has the users preloaded image displayed as a backdrop to the event monitor menu 1100 as previously. The event monitor menu comprises a dialogue box having data entry areas 1101-1103, each having a drop down menu, for selecting items on the computer platform such as a user file, a driver, or an application. In general, any physical or logical component of the computer platform which gives rise to event data when events occur on that component can be selected by the trusted component. For ease of description, in the following, selections will be

described primarily in relation to data files, application programs and drivers, although it will be appreciated that the general methods and principles described herein are applicable to the general set of components and facilities of the computer platform. By activating the drop down menu on each of selection boxes 1101-1103, there is listed a corresponding respective list of data files, drivers, or applications which are present on the computer platform. A user may select any of these files and/or applications and/or drivers by activating the pointing device on the selected icon from the drop down menu in conventional manner in steps 904, 905, 906. Additionally, the event monitor menu comprises an event select menu 1104. The event select menu lists a plurality of event types which can be monitored by the event logger 702 within the trusted component, for the file, application or driver which is selected in selection boxes 1101, 1102, 1103 respectively. Types of event which can be monitored include events in the set: file copied - the event of a selected file being copied by an application or user; file saved - the event of whether a specified file is saved by an application or user; file renamed - the event of whether a file has been renamed by an application or user; file opened - the event of whether a file is opened by an application or user; file overwritten - the event of whether data within a file has been overwritten; file read - the event of whether data in a file has been read by any user, application or other entity; file modified - the event of whether data in a file has been modified by a user, application or other entity; file printed - the event of whether a file has been sent to a print port of the computer entity; driver used - whether a particular driver has been used by any application or file; driver reconfigured - the event of whether a driver has been reconfigured; modem used - subset of the driver used event, applying to whether a modem has been used or not; disk drive used - the event of whether a disk drive has been used in any way, either written or read; application opened - the event of whether an application has been opened; and application closed - the event of whether an application has been closed. Once the user has selected the application, driver or file and the events to be monitored in dialog box 1100, the user activates the confirmation key 104, which is confirmed by confirmation key icon 1105 visually altering, in order to activate a monitoring session. A monitoring session can only be activated by use of the dialog box 1100, having the user's image 1001 from the user's smart card display thereon, and by independently pressing confirmation key 104. Display of the image 1001 on the monitor 100, enables the user to have confidence that the trusted component is generating the dialog box. Pressing the confirmation key 104 by the user, which is directly input into trusted component 202 independently of the computer platform gives direct confirmation to the trusted component that the user, and not some other entity, e. g. a virus or the like is activating the monitoring session.

[0061] The user may also specify a monitoring period by entering a start time and date and a stop time and date in data entry window 1106. Alternatively, where a single event on a specified entity is to be monitored, the user can specify monitoring of that event only by confirming with pointing device 105 in first event only selection box 1107.

[0062] Two modes of operation will now be described, in the first mode of operation, continuous event monitoring of specified entities over a user specified period occurs. In the second mode of operation, continuous monitoring of a specified entity occurs until a user specified event has happened, or until a user specified period for monitoring that user specified event has elapsed.

[0063] In Fig. 12 herein, there is illustrated a procedure for continuous monitoring of a specified logical or physical entity over a user specified monitoring period.

[0064] Referring to Fig. 12 herein, there is illustrated schematically process steps operated by trusted component 202 in response to a user input to start an event monitoring session as described with reference to figs. 8 to 11 herein before. In step 1200, display interface 701 receives commands from the user via the dialogue boxes which are input using pointing device 105, keyboard 101 via data bus 304 and via communications interface 700 of the trusted component. The event logger 702 instructs agent 511 in user space to commence event monitoring. The instructions comprising event logger 702 are stored within a memory area resident within the trusted component 202. Additionally, event logger 702 is also executed within a memory area in the trusted component. In contrast, whilst the instructions comprising agent 511 are stored inside the trusted components 202 in a form suitable for execution on the host processor ie in CPU native programs area 403 of the trust component, agent 511 is executed within untrusted user space ie outside of the trusted component 202. Agent 511 receives details of the file, application and/or drivers to be monitored from event logger 702. In step 1200, agent 511 receives a series of event data from the logical entity (eg file, application or driver) specified. Such monitoring is a continuous process, and agent 511 may perform step 1200 by periodically reading a data file in which such event data is automatically stored by the operating system (for example in the Microsoft windows 4.O™ operating system which contains the facility for logging events on a file). However, in order to maximize security, it is preferable the agent 511 periodically gathers event data itself by interrogating the file, application or driver directly to elicit a response. In step 1201, the collected data concerning the events of entity are reported directly to the trusted component 202, which then stores them in a trusted memory area in step 1202. In step 1203, the event logger checks whether the user specified predetermined monitoring period from the start of the event monitoring session has elapsed. If the event monitoring session period has not yet elapsed, event logger 702 continues to await further events on the specified files, applications or drivers supported by

the agent 511, which steps through steps 1200 - 1202 as previously until the predetermined user specified period has elapsed in step 1203. In step 1204, the trusted component takes the content of the event data stored in trusted memory and applies cryptographic function 703 to the event log to provide a secure event log file. The process of securing the event log file as described herein before is such that the secured file has at least the properties of:

- Authentication - an authorised user or program should be able to correctly ascertain the origin of the event log file;
- Integrity - It should be possible to verify that the event log file has not been modified by an unauthorised individual or program.

[0065] Optionally, the secured file should have the property of confidentiality - unauthorised users or programs should not be able to access the information contained within the event log file; and the property of non-repudiation - proper authentication of data cannot later be falsely denied.

[0066] The trusted component in step 1205 writes the secure event log file to a memory device. The memory device may either be in trusted space, or in user space. For example the secure event log file may be stored in a user accessible portion of a hard disk drive 203.

[0067] By providing a secure event log file containing data describing a plurality of events which have occurred on a specified file, application or driver, a user reading the file can be confident that the data in the file has been written by the trusted component and has not been corrupted. Any corruption to the data are immediately evident. In the best mode herein, securing of the event log file is made by applying a chaining algorithm which chains arbitrary chunks of data as is known in the art. In such chaining processes, the output of a previous encryption process is used to initialize a next encryption process. The amounts of data in each encrypted data block are of arbitrary length, rather than being a single plain text block. Details of such chaining algorithms which are known in the art can be found in "Handbook of Applied Cryptography", Menezes Vanoorschot, Vanstone, on page 229. The key used during the chaining process is one stored within the trusted component 202, preferably the private signature key of the trusted component. The validity of the secured event log can then readily be confirmed by any entity possessing the public signature key of the trusted component. Such methods are well known to those skilled in the art of information security.

[0068] Event data is preferably gathered by the use of additional device drivers. NT is designed so that additional device drivers may be inserted between existing device drivers. It is therefore possible to design and insert drivers that trap access to files, applications, and other device drivers, and provide details of the interactions as event data. Information on the design and use of device drivers may be found, for example, in the 'The Windows NT Device Driver Book' (author A.Baker, published by Prentice Hall). Also, commercial companies such as 'BlueWater Systems' offer device driver toolkits.

[0069] Referring to Fig. 13 herein, there is illustrated a set of process steps applied by the trusted component and agent 511 for monitoring one off special events specified by the user by data entry through dialogue boxes as described herein before. Details of special events to be monitored are specified by the user in step 1300. Details of the particular entity, eg a file application or driver to be monitored are entered in step 1301. In step 1302, details of the event types and entity to be monitored are sent to the agent 511 from the trusted component. The agent then proceeds to continuously monitor for the events on that particular specified entity in step 1303. Periodically, it is checked whether any event has occurred in step 1304 by the agent, and if no event has yet occurred, the agent continues in step 1303 to monitor the specified entity. When an event has occurred, in step 1305 details are passed back to the trusted component in step 1305. The trusted component then applies a cryptographic function to the event data to provide secure event data in step 1306, and in step 1307 writes the secure event data to a memory area either in trusted space or in user space as herein before described with reference to Fig. 12.

[0070] The secure event data is a log that can be used, for example, for auditing. An investigator can inspect the log comprised of the secure event data. That investigator can use standard cryptographic techniques to verify the integrity of the event data, and that it is complete. The investigator can then construct a history of the platform. This is useful for investigating attacks on the platform, or alleged improper use of the platform. The event data has been gathered by an impartial entity (the trusted component 202) whose behavior cannot be modified by a user or unilaterally by the owner of the platform. Hence the event log serves as an honest record of activities within the platform. The event log can be published as a report or automatically interpreted by, for example, a computer program that is outside the scope of this invention.

[0071] Types of event data which may be stored in the event log include the following. The following lists should be regarded as a non-exhaustive, and in other embodiments of the present invention common variations as will be recognized by those skilled in the art may be made: a time of an event occurring; a date of an event occurring, whether or not a password has been used, if a file is copied, a destination to which the file has been copied to; if a file has been operated on, a size of the file in megabytes; a duration for which a file was open; a duration over which an application has been online; a duration of which a driver has been online; an internet address to which a file has been copied, or to which a driver has accessed, or to which an application has ad-

dressed; a network address to which a file has been copied, to which an application has addressed, or to which a driver has corresponded with.

[0072] The event data stored in the event log may be physically stored in a data file either on the platform or in the trusted component. The event log data is secured using a chaining function, such that a first secured event data is used to secure a second secured event data, a second secured event data is used to secure a third event data, etc so any changes to the chain of data are apparent.

[0073] In addition to providing the secured event log data, the trusted component may also compile a report of events. The report may be displayed on monitor 100. Items which may form the content of a report include the events as specified in the event log above, together with the following: time of an event, date of an event, whether or not a password was used, a destination of the file it is copied to, a size of a file (in megabytes), a duration a file or application has been open, a duration over which a driver has been online, a duration over which a driver has been used, a port which has been used, an internet address which has been communicated with, a network address which has been communicated with.

[0074] Agent 511 performs event monitoring operations on behalf of trusted component 202, however whereas trusted component 202 is resident in a trusted space 513, agent 511 must operate in the user space of the computer platform. Because the agent 511 is in an inherently less secure environment than the trusted space 513, there is the possibility that agent 511 may become compromised by hostile attack to the computer platform through a virus or the like. The trusted component deals with the possibility of such hostile attack by either of two mechanisms. Firstly, in an alternative embodiment the agent 511 may be solely resident within trusted component 202. All operations performed by agent 511 are performed from within trusted user space 513 by the monitoring code component 600 operating through the trusted components' communications interface 700 to collect event data. However, a disadvantage of this approach is that since agent 511 does not exist, it cannot act as a buffer between trusted component 202 and the remaining user space 504.

[0075] On the other hand, the code comprising agent 511 can be stored within trusted space in a trusted memory area of trusted component 202, and periodically "launched" into user space 504. That is to say, when a monitoring session is to begin, the agent can be downloaded from the trusted component into the user space or kernel space on the computer platform, where it then resides, performing its continuous monitoring functions. In this second method, which is the best mode contemplated by the inventors, to reduce the risk of any compromises of agent 511 remaining undetected, the trusted component can either re-launch the complete agent from the secure memory area in trusted space into the user space at periodic intervals, and/or can periodically monitor the agent 511 in user space to make sure that it is responding correctly to periodic interrogation by the trusted component.

[0076] Where the agent 511 is launched into user space from its permanent residence in trusted space, this is effected by copying code comprising the agent from the trusted component onto the computer platform. Where a monitoring session has a finite monitoring period specified by a user, the period over which the agent 511 exists in user space can be configured to coincide with the period of the monitoring session. That is to say the agent exists for the duration of the monitoring session only, and once the monitoring session is over, the agent can be deleted from user/kernel space. To start a new monitoring session for a new set of events and/or entities, a new agent can be launched into user space for the duration of that monitoring session.

[0077] During the monitoring session, which may extend over a prolonged period of days or months as specified by a user, the trusted component monitors the agent itself periodically.

[0078] Referring to Fig. 14 herein, there is illustrated schematically process steps carried out by trusted component 202 and agent 511 on the computer platform for launching the agent 511 which is downloaded from trusted space to user space, and in which the trusted component monitors the agent 511 once set up and running on the computer platform.

[0079] In step 1400, native code comprising the agent 511 stored in the trusted components secure memory area is downloaded onto the computer platform by the computer platform reading the agent code directly from the trusted component in step 1401. In step 1402, the data processor on the computer platform commences execution of the native agent code resident in user space on the computer platform. The agent continues to operate as described herein before continuously in step 1403. Meanwhile, trusted component 202 generates a nonce challenge message in step 1404 after a suitable selected interval, and sends this nonce to the agent which receives it in step 1405. The nonce may comprise a random bit sequence generated by the trusted component. The purpose of the nonce is to allow the trusted component to check that the agent is still there and is still operating. If the nonce is not returned by the agent, then the trusted component knows that the agent has ceased to operate and/or has been compromised. In step 1407 the agent signs the nonce and in step 1408 the agent sends the signed nonce back to the trusted component. The trusted component receives the signed nonce in step 1409 and then repeats step 1404 sending a new nonce after a pre-selected period. If after a predetermined wait period 1406, commencing when the nonce was sent to the agent in step 1404, the trusted component has not received a nonce returned from the agent, then in step 1410 the trusted component generates an alarm signal which may result in a display on the monitor showing that the agent 511 is incorrectly op-

erating, and that file monitoring operations may have been compromised.

[0080] In a second embodiment, trusted component 202 may operate to gather information about the use of data and platform resources with programs using utilities and functions provided by the operating system resident on the computer platform. This information may include access rights, file usage, application usage, memory (RAM) utilization, memory (hard disk) utilization, and main processor instruction cycle allocation statistics.

[0081] The prior patent application 'Trusted Computing Platform' describes a method whereby the trusted component cooperates with other entities and reports to them the values of integrity metrics measured by the trusted component. Those other entities then compare the measured metrics with the proper values that are contained in a digital certificate published by a trusted third party. That prior patent application gives an example of a static metric - a digest of the platform's BIOS memory. The measurements made by the method of this application may also be reported as integrity metrics, but because they are potentially always changing, they are called dynamic integrity metrics - a measured value may be different now from the value measured a few seconds previously. Entities must repeatedly request the current value of a measured dynamic metric. For example one integrity metric, according to the best mode described herein, comprises a Boolean value which indicates whether an event which has occurred is apparently incompatible with a policy governing access to data. For example such a Boolean would be TRUE if a mobile software such as a Java applet wrote over files in the user space, even though the mobile software did not have write permission to those files.

[0082] Another integrity metric comprises a Boolean value which indicates that unusual behavior has been detected. Such unusual behavior may not necessarily indicate that the computer platform has become unsafe, but may suggest caution in use of the computer platform. Prudent entities communicating with the computer platform may choose not to process very sensitive data on that platform if the second integrity metric indicates that unusual behavior has been detected. Unusual behavior is difficult to accurately define, unless a platform is used to do repetitive operations. In the best mode herein, unusual data may be defined and monitored for by the trusted component as being behavior of a resource on the computer platform which is outside a predetermined number of standard deviations of a historical mean measurement of behavior compiled over a predetermined period. For example where a data file has historically over a pre-determined period had a size within a particular range, eg 140- 180 megabytes, if the file size increases dramatically, eg to 500 megabytes, and outside a pre-determined number of standard deviations which can be preset, then the second integrity metric Boolean value may change state to a true state, indicating unusual behavior.

[0083] As a further example, if an application, eg a word processing application, has a history of saving data files with a frequency in a predetermined range, for example in the range of 1 to 10 saves per day, and the application changes behavior significantly, eg saving 100 saves per day, then a Boolean metric for monitoring that parameter may trigger to a true state.

[0084] Of course, as previously mentioned, it may be that the trusted component takes a proactive role in reporting urgent events, instead of waiting to be polled by an integrity challenge. Events can be matched inside the trusted component 202 with policy rules stored inside the trusted component. If an event breaches a rule that the policy considers to be crucial, the trusted component 202 can immediately send an alarm indication message to a relevant entity, and/or display an emergency message to the user on the monitor 100 using the style of dialog box indicated in Figures 10 and 11.

Claims

1. A computer entity comprising:

   a computer platform comprising a data processor and at least one memory device; and

   a trusted component, said trusted component comprising a data processor and at least one memory device;

   wherein said data processor and said memory of said trusted component are physically and logically distinct from said data processor and memory of said computer platform; and

   means for monitoring a plurality of events occurring on said computer platform.

2. The computer entity as claimed in claim 1, wherein said monitoring means comprises a software agent operating on said computer platform, for monitoring at least one event occurring on said computer platform, and reporting said event to said trusted component.

3. The computer entity as claimed in claim 2, wherein said software agent comprises a set of program code normally resident in said memory device of said trusted component, said code being transferred into said computer platform for performing monitoring functions on said computer platform.

4. The computer entity as claimed in claim 1, where said trusted component comprises an event logging component for receiving data describing a plurality of events occurring on said computer platform, and

compiling said event data into secure event data.

5. The computer entity as claimed in claim 4, wherein said event logging component comprises means for applying a chaining function to said event data to produce said secure event data.

6. The computer entity as claimed in claim 1, further comprising a display interface for generating an interactive display comprising:

means for selecting an entity of said computer platform to be monitored; and

means for selecting at least one event to be monitored.

7. The computer entity as claimed in claim 1, further comprising prediction means for predicting a future value of at least one selected parameter.

8. The computer entity as claimed in claim 1, further comprising a confirmation key means connected to said trusted component, and independent of said computer platform, for confirming to said trusted component an authorisation signal of a user.

9. The computer entity as claimed in claim 1, wherein logical entities to be monitored are selected from the set:

at least one data file;

at least one application;

at least one driver component.

10. A computer entity comprising:

a computer platform having a first data processor and a first memory device; and

a trusted monitoring component comprising a second data processor and a second memory device, wherein

said trusted monitoring component stores an agent program resident in said second memory area, said agent program arranged to be copied to said first memory area for performing functions on behalf of said trusted component, under control of said first data processor.

11. A computer entity comprising:

a computer platform comprising a first data processor and a first memory device;

a trusted monitoring component comprising a second data processor and a second memory device;

a first computer program resident in said first memory area and operating said first data processor, said first computer program reporting back events concerning operation of said computer platform to said trusted monitoring component; and

a second computer program said second computer program resident in said second memory area of said trusted component, said second program operating to monitor an integrity of said first program.

12. The computer entity as claimed in claim 11, wherein said computer program monitors an integrity of said first computer program by sending to said first computer program a plurality of interrogation messages, and monitoring a reply to said interrogation messages made by said first computer program.

13. The computer entity as claimed in claim 12, wherein a said interrogation message is sent in a first format, and returned in a second format, wherein said second format is a secure format.

14. A method of monitoring a computer platform comprising a first data processor and a first memory means, said method comprising the steps of:

reading event data describing events occurring on at least one logical or physical entity comprising said computer platform;

securing said event data in a second data processing means having an associated second memory area, said second data processing means, said second memory area being physically and logically distinct from said first data processing means and said first memory area, such that said secure event data cannot be altered without such alteration being apparent.

15. The method as claimed in claim 14, where a said event to be monitored is selected from the set of events:

copying of a data file;

saving a data file;

renaming a data file;

opening a data file;

overwriting a data file;

modifying a data file;

printing a data file;

activating a driver device;

reconfiguring a driver device;

writing to a hard disk drive;

reading a hard disk drive;

opening an application;

closing an application.

16. The method as claimed in claim 14, wherein a said entity to be monitored is selected from the set:

at least one data file stored on said computer platform;

a driver device of said computer platform;

an application program resident on said computer platform.

17. The method as claimed in claim 14, wherein said step of monitoring said entity comprises continuously monitoring a said entity over a pre-selected time period.

18. The method as claimed in claim 14, wherein said step of monitoring said entity comprises:
monitoring said entity until such time as a pre-selected event occurs on said entity.

19. The method as claimed in claim 14, wherein said step of monitoring said entity comprises:
monitoring a said entity for a selected event, until a predetermined time period has elapsed.

20. A method of monitoring a computer platform comprising a first data processing means and a first memory means, said method comprising the steps of:

generating an interactive display for selecting at least one entity comprising said computer platform;

generating a display of events which can be monitored;

generating a display of entities of said computer platform;

selecting at least one said entity;

selecting at least one said event; and

monitoring a said entity for a said event.

21. A method of monitoring a computer platform comprising a first data processing means and first memory means, said method comprising the steps of:

storing a monitoring program in a second memory area, said second memory area being physically and logically distinct from said first memory area;

transferring said monitoring program from said second memory area to said first memory area;

monitoring at least one entity of said computer platform from within said computer platform; and

reporting an event data from said monitoring program to said second data processor.

22. A method of monitoring a computer platform comprising a first data processing and a first memory means, said method comprising the steps of;

monitoring at least one entity comprising said computer platform from within said computer platform;

generating an event data describing a plurality of events occurring on said computer platform;

reporting said event data to a second data processing means having an associated second memory means; and

processing said event data into an secure format.

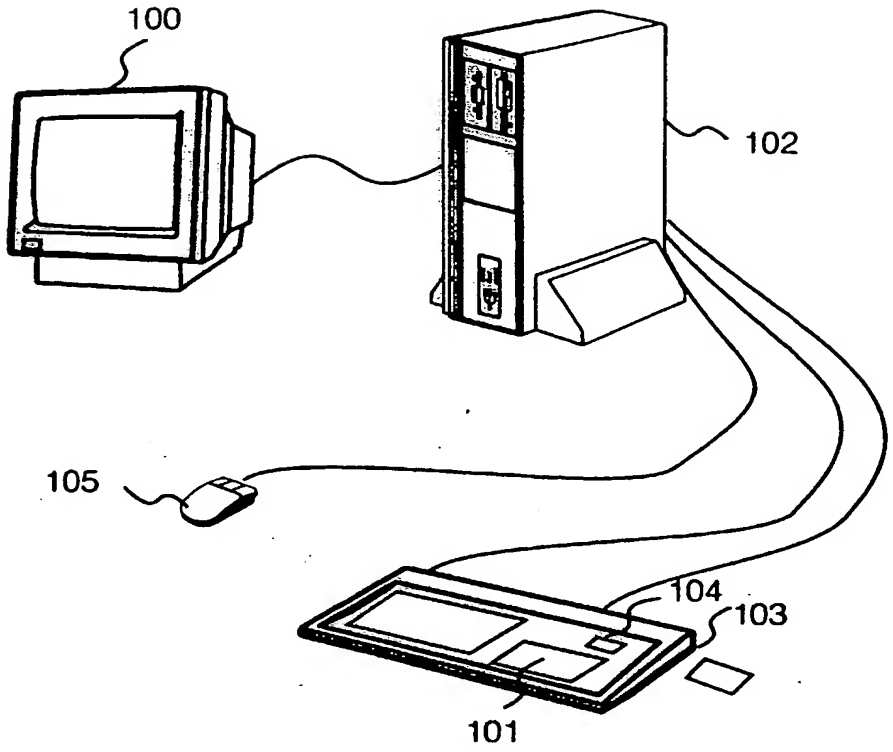

Fig. 1

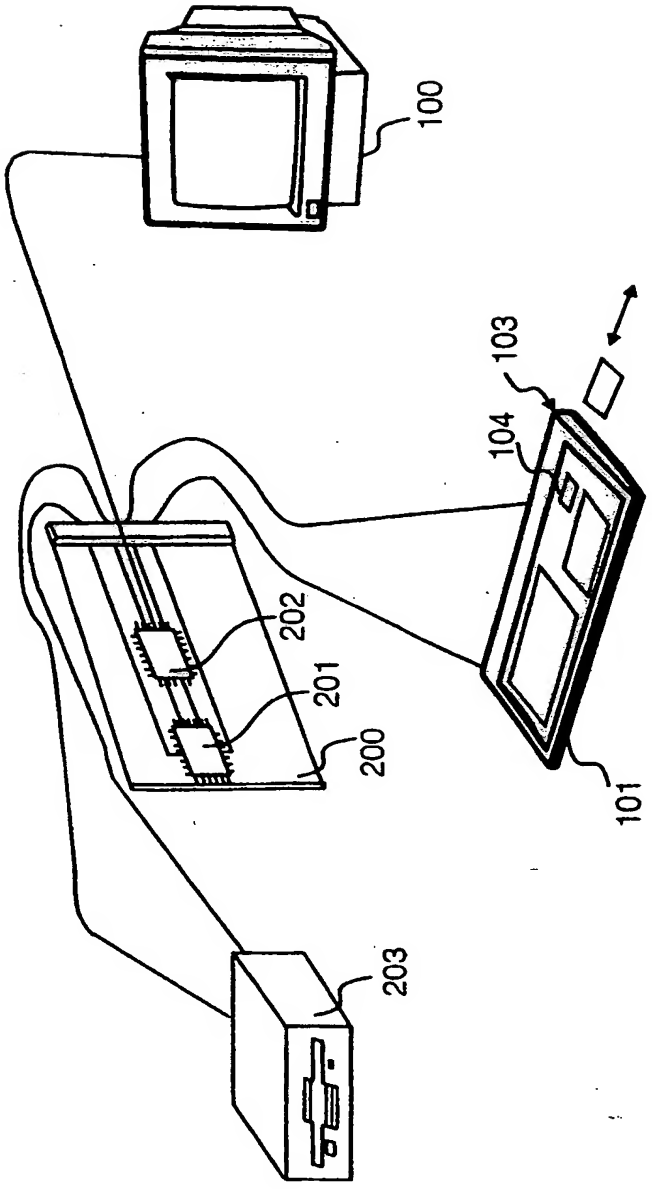

Fig. 2

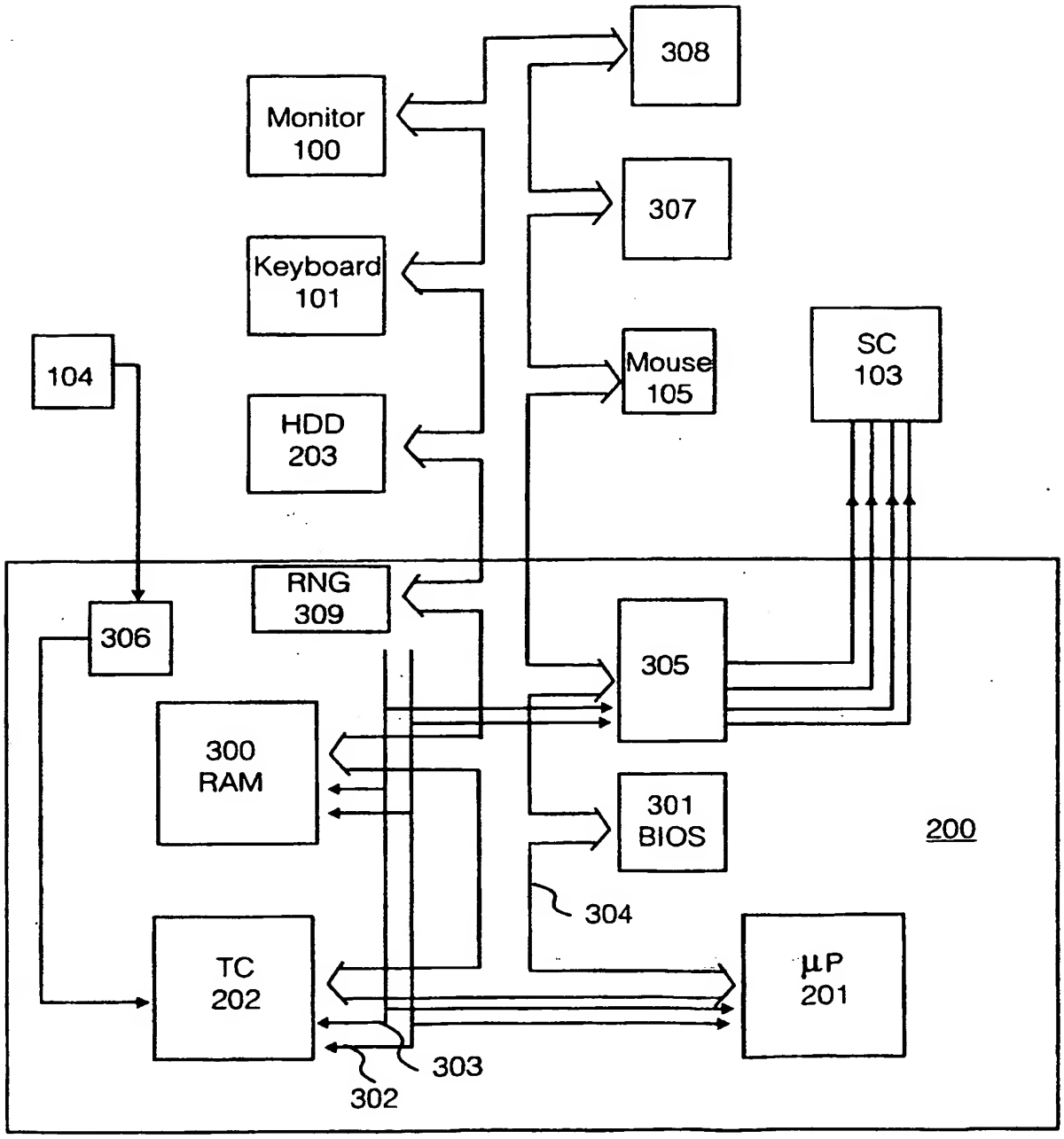
308
Monitor
100
307
Keyboard
101
104
Mouse
105
SC
103
HDD
203
RNG
309
306
305
300
RAM
301
BIOS
200
304
TC
202
µP
201
303
302

Fig. 3

This Page Blank (uspto)

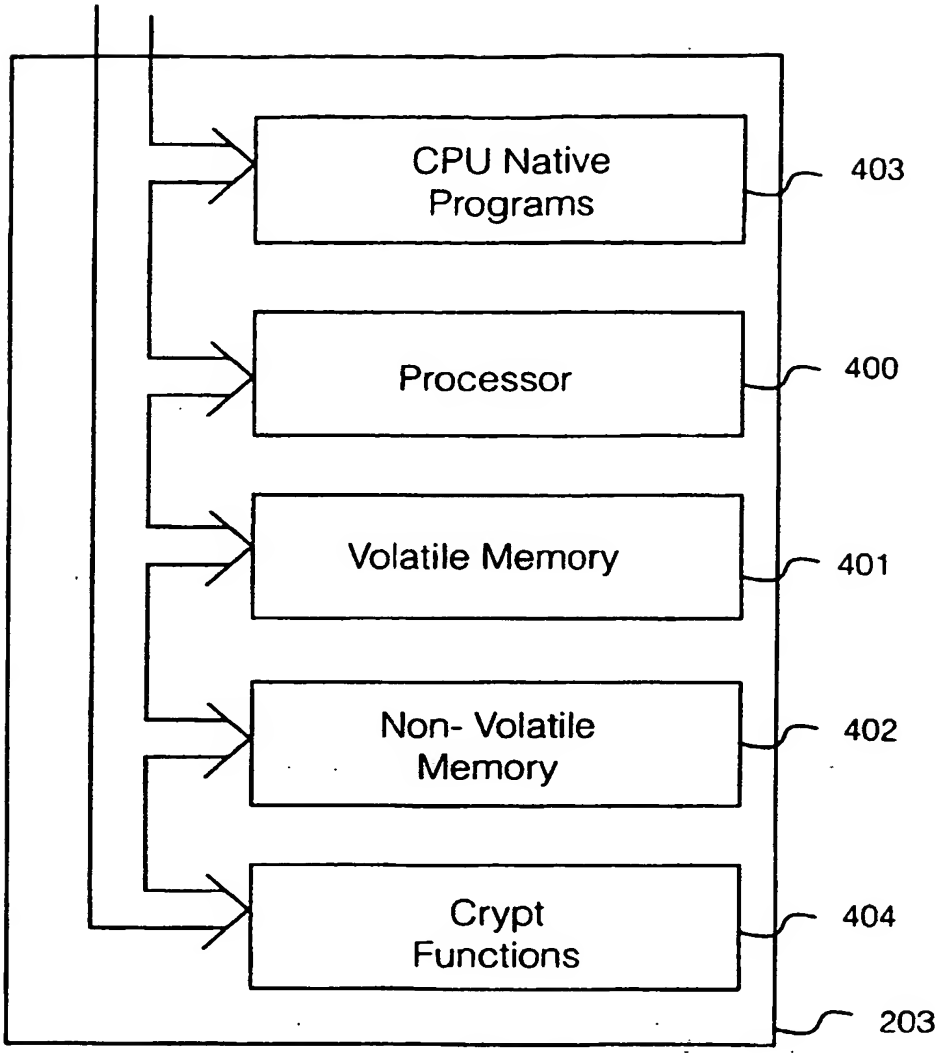
CPU Native Programs
403
Processor
400
Volatile Memory
401
Non- Volatile Memory
402
Crypt Functions
404
203

Fig. 4

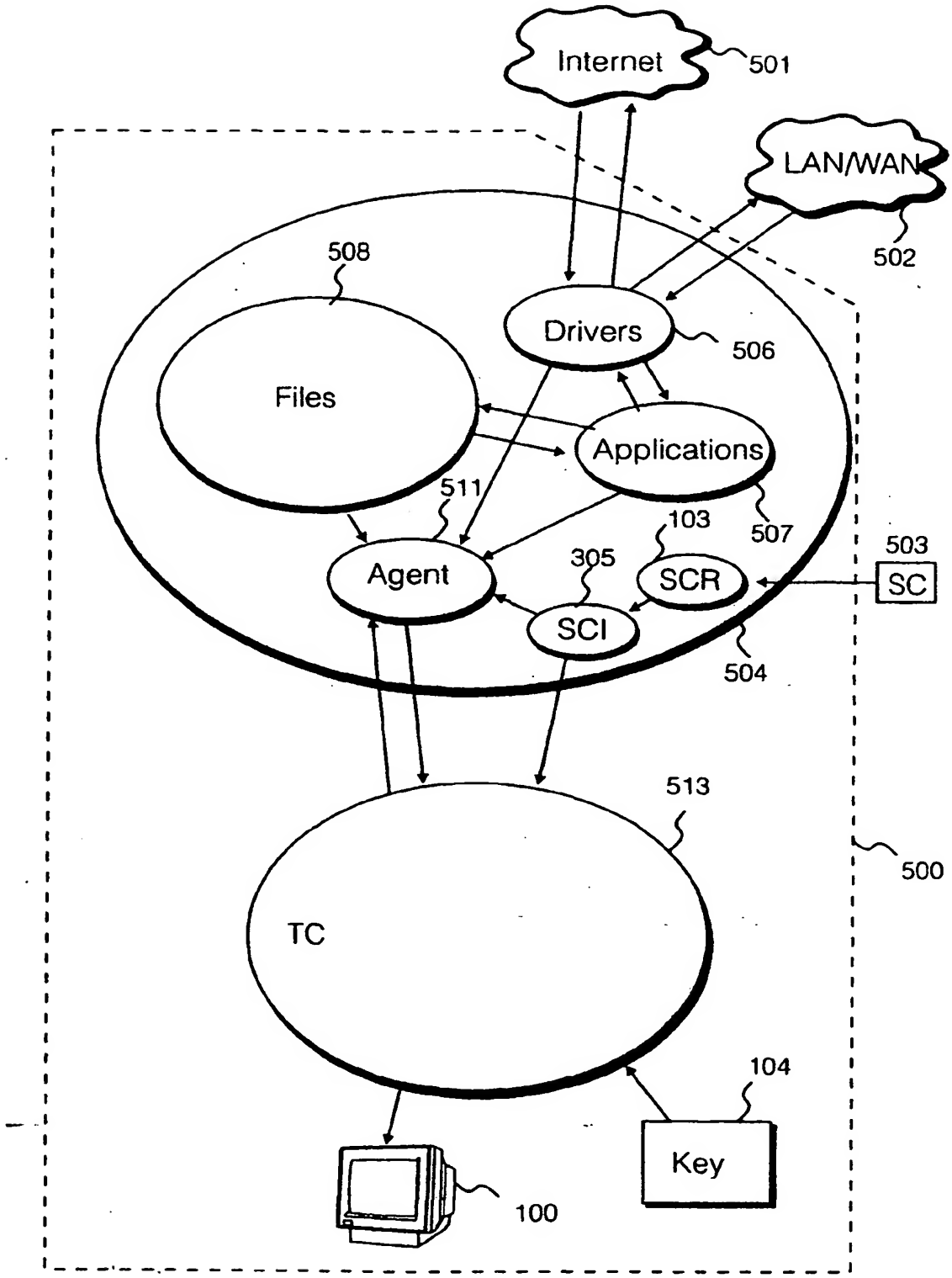
Internet
501
LAN/WAN
502
508
Drivers
506
Files
Applications
511
103
507
503
305
Agent
SCR
SC
SCI
504
513
500
TC
104
Key
100

Fig. 5

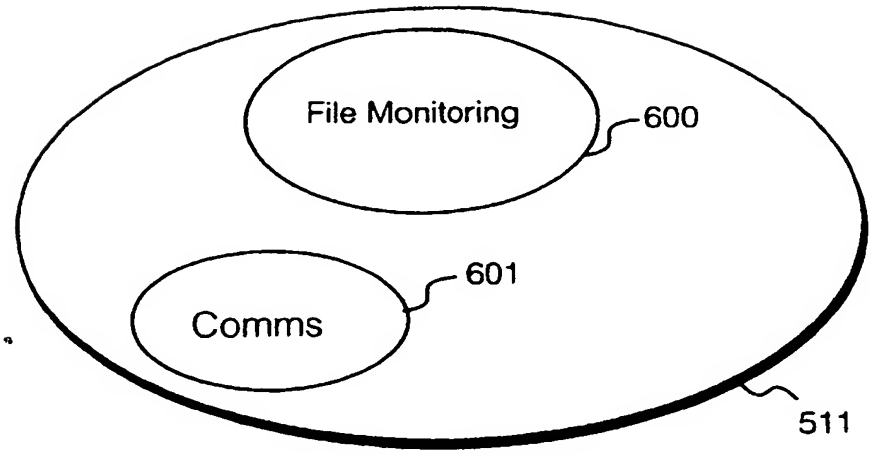
File Monitoring
600
601
Comms
511

Fig. 6

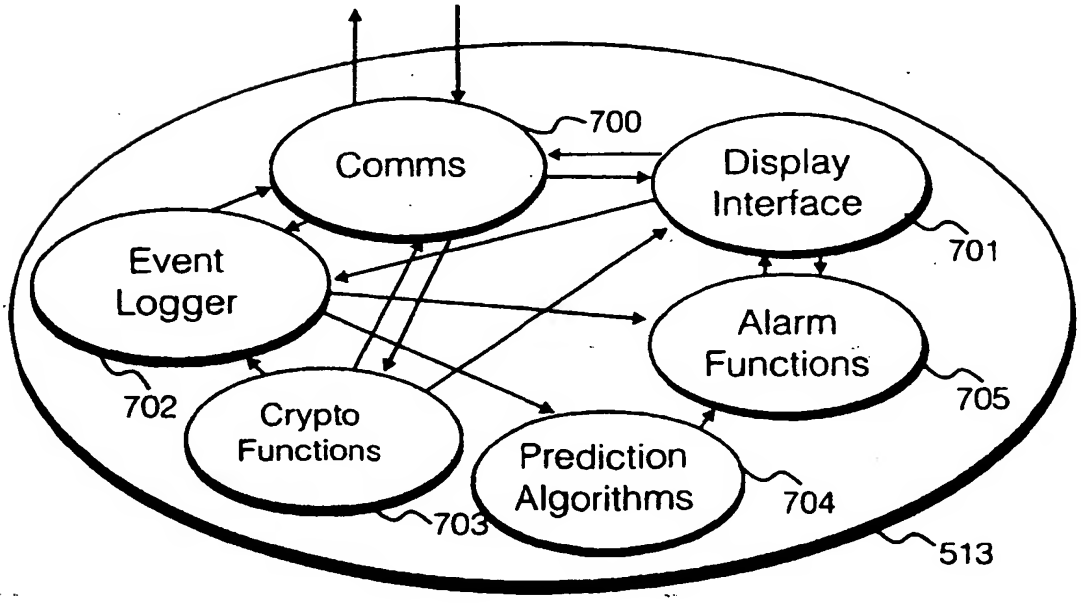
700
Comms
Display Interface
701
Event Logger
Alarm Functions
705
702
Crypto Functions
Prediction Algorithms
704
703
513

Fig. 7

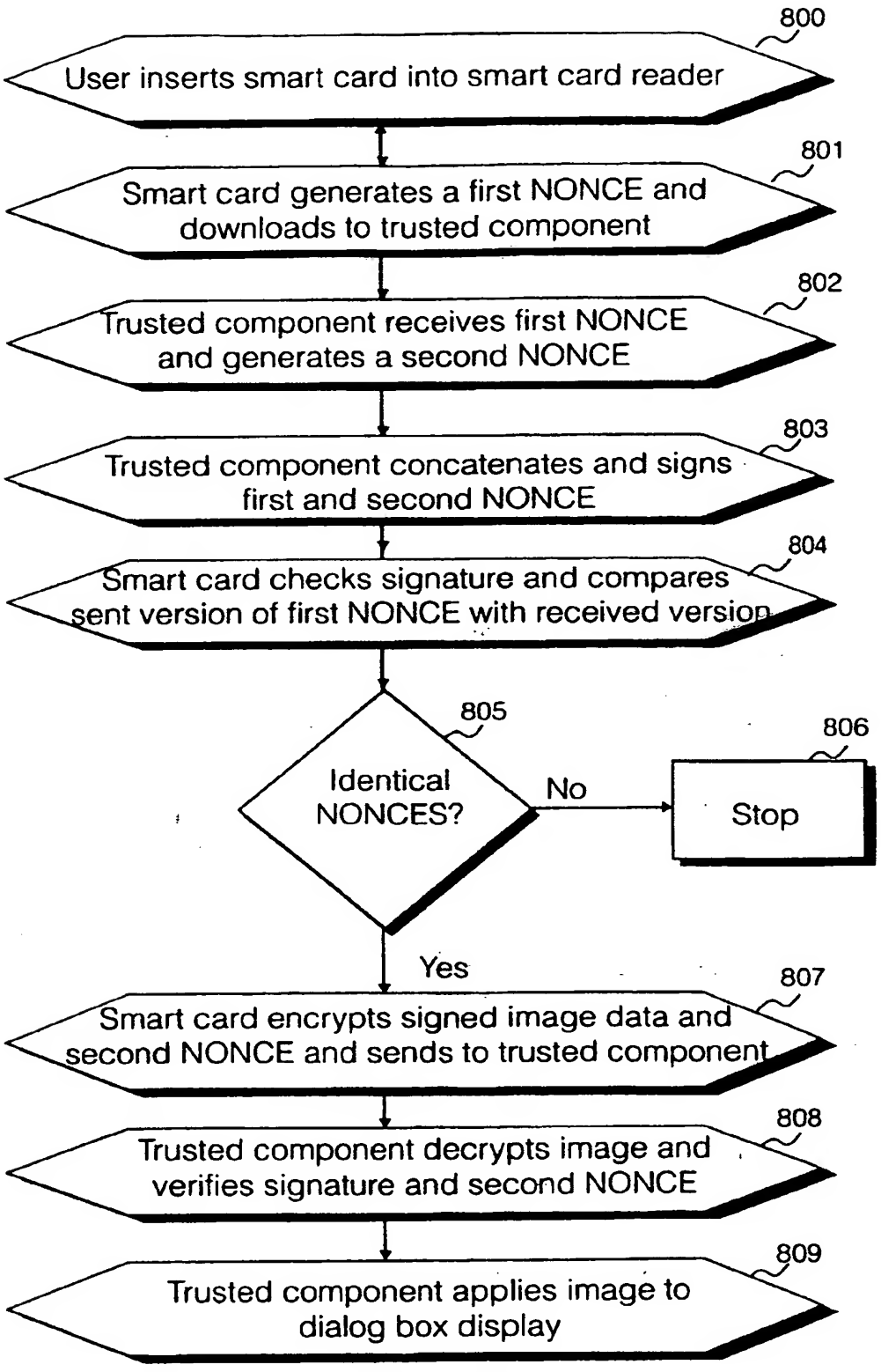
800
User inserts smart card into smart card reader
801
Smart card generates a first NONCE and downloads to trusted component
802
Trusted component receives first NONCE and generates a second NONCE
803
Trusted component concatenates and signs first and second NONCE
804
Smart card checks signature and compares sent version of first NONCE with received version
805
Identical NONCES?
No
806
Stop
Yes
807
Smart card encrypts signed image data and second NONCE and sends to trusted component
808
Trusted component decrypts image and verifies signature and second NONCE
809
Trusted component applies image to dialog box display

Fig. 8

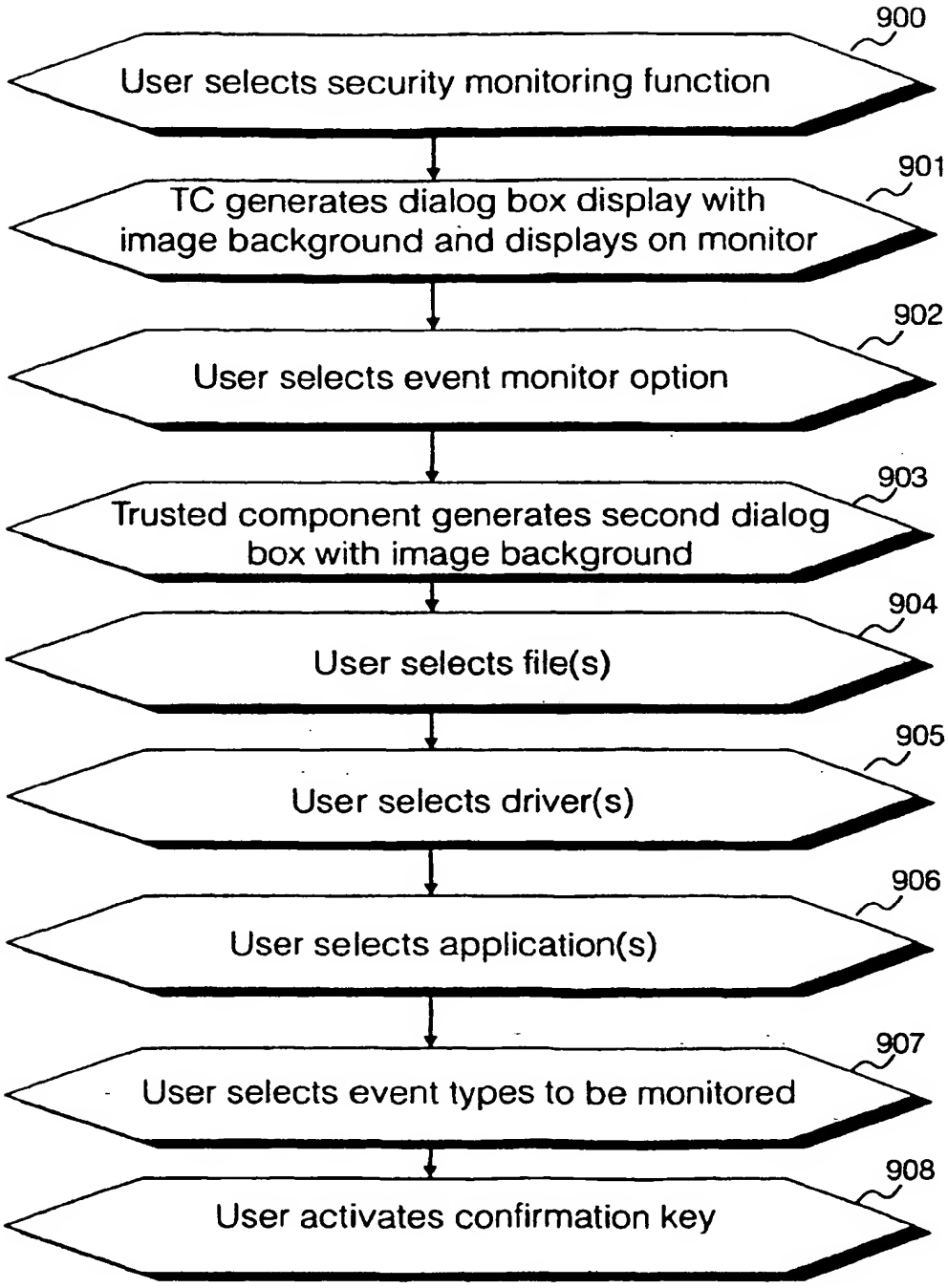
900
User selects security monitoring function
901
TC generates dialog box display with image background and displays on monitor
902
User selects event monitor option
903
Trusted component generates second dialog box with image background
904
User selects file(s)
905
User selects driver(s)
906
User selects application(s)
907
User selects event types to be monitored
908
User activates confirmation key

Fig. 9

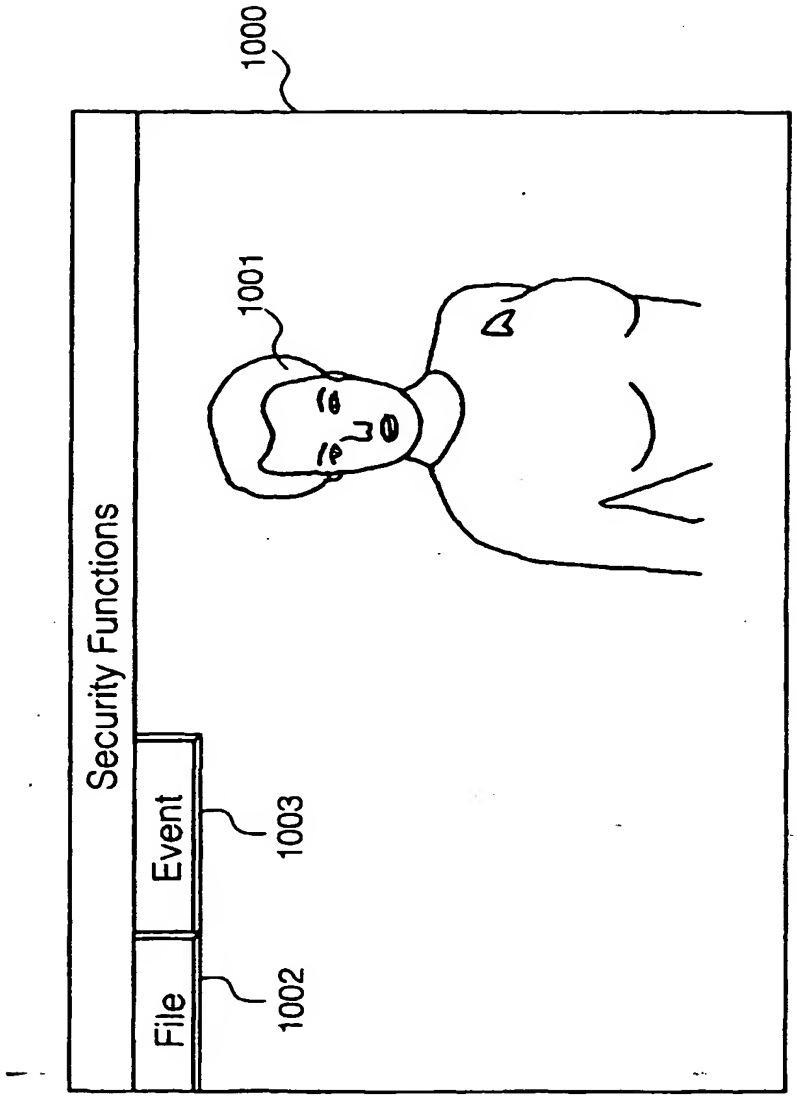

Fig. 10

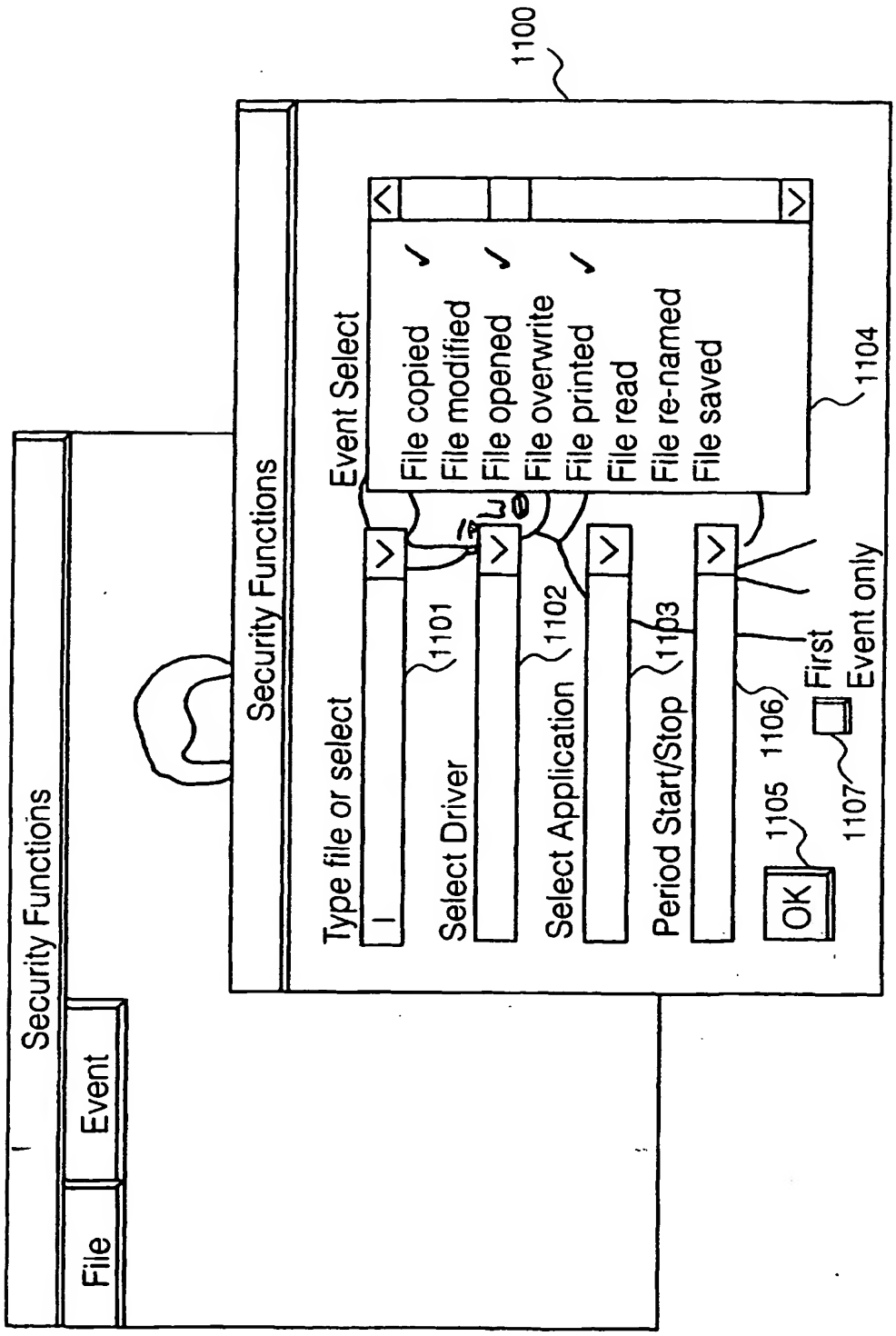
1100
Security Functions
Event
File
Security Functions
Event Select
File copied
File modified
File opened
File overwrite
File printed
File read
File re-named
File saved
1104
Type file or select
1101
Select Driver
1102
Select Application
1103
Period Start/Stop
1106
First
Event only
1105
OK
1107

Fig. 11

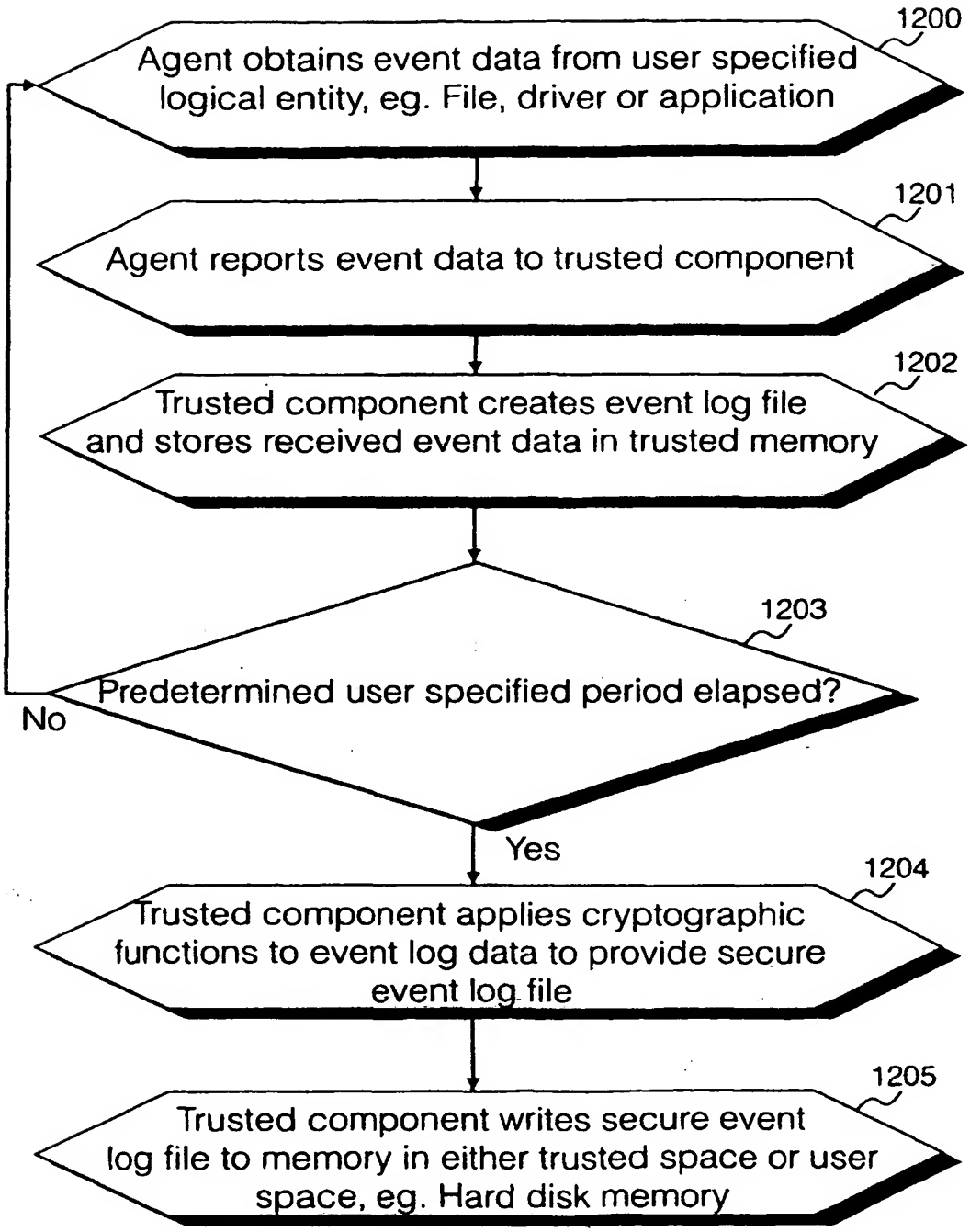
1200
Agent obtains event data from user specified logical entity, eg. File, driver or application
1201
Agent reports event data to trusted component
1202
Trusted component creates event log file and stores received event data in trusted memory
1203
Predetermined user specified period elapsed?
No
Yes
1204
Trusted component applies cryptographic functions to event log data to provide secure event log file
1205
Trusted component writes secure event log file to memory in either trusted space or user space, eg. Hard disk memory

Fig. 12

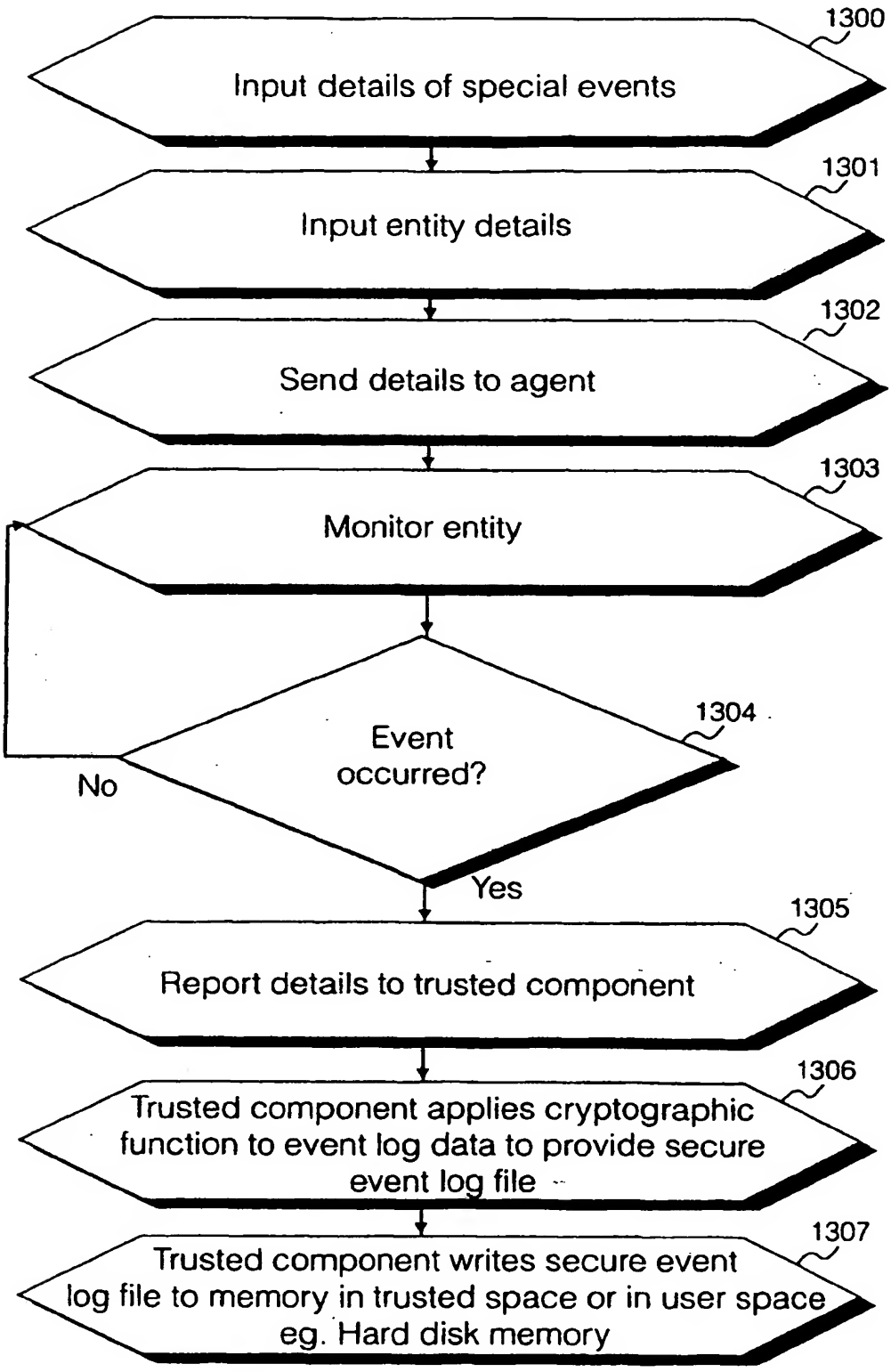
1300
Input details of special events
1301
Input entity details
1302
Send details to agent
1303
Monitor entity
1304
Event occurred?
No
Yes
1305
Report details to trusted component
1306
Trusted component applies cryptographic function to event log data to provide secure event log file
1307
Trusted component writes secure event log file to memory in trusted space or in user space eg. Hard disk memory

Fig. 13

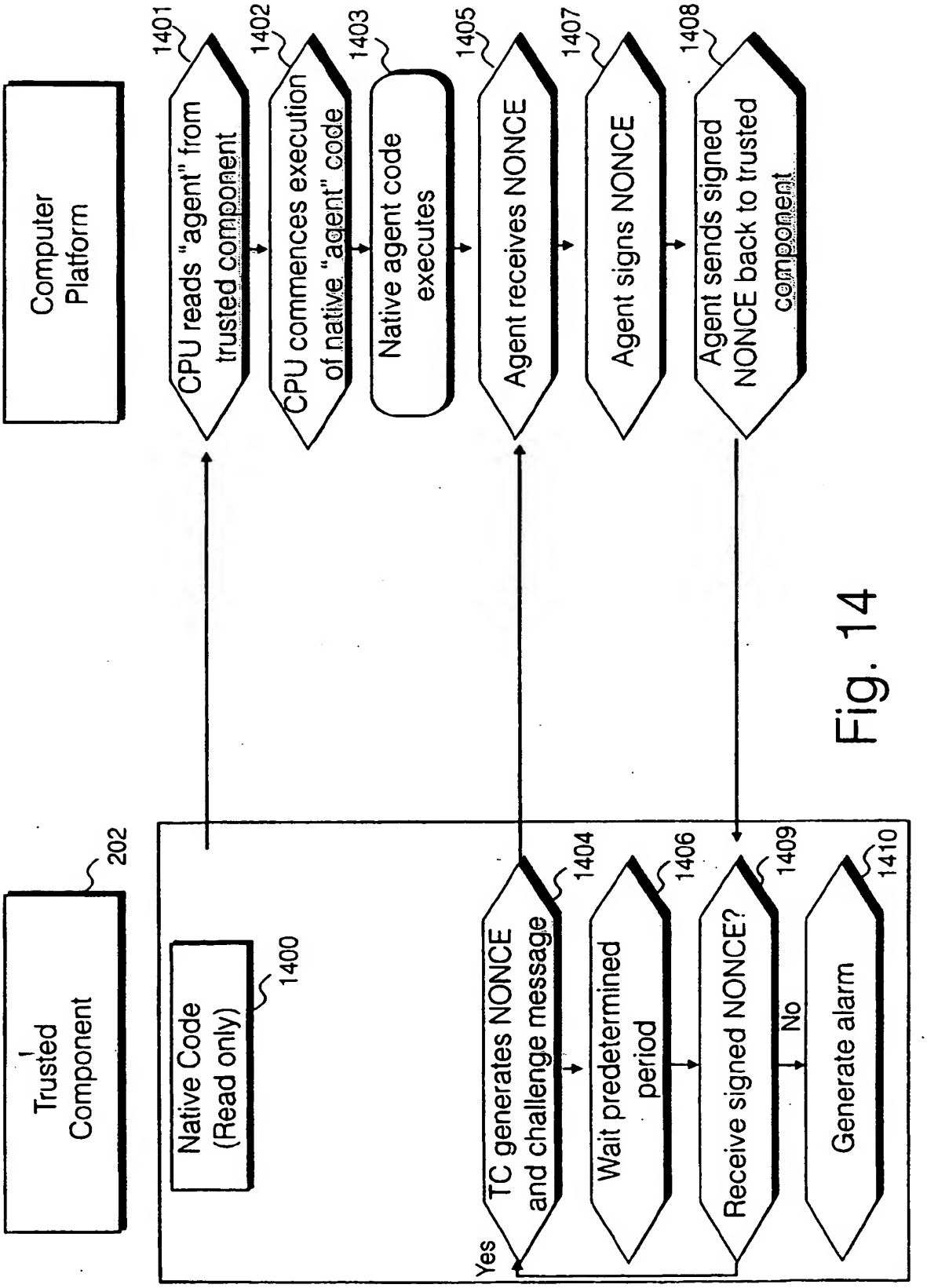
Computer Platform
1401
CPU reads "agent" from trusted component
1402
CPU commences execution of native "agent" code
1403
Native agent code executes
1405
Agent receives NONCE
1407
Agent signs NONCE
1408
Agent sends signed NONCE back to trusted component
Trusted Component
202
Native Code (Read only)
1400
1404
TC generates NONCE and challenge message
1406
Wait predetermined period
1409
Receive signed NONCE?
No
1410
Generate alarm
Yes

Fig. 14

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 30 4165

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	WO 98 45778 A (ZUTA MARC) 15 October 1998 (1998-10-15) * abstract; figure 1 * * page 16, line 1 - page 20, last line * * page 26, line 1 - page 32, last line *	1,14-16	G06F1/00
Y		2,3,10, 11,17-22	
Y	US 5 404 532 A (ALLEN WADE C ET AL) 4 April 1995 (1995-04-04) * the whole document *	2,3,10, 11	
Y	WO 95 27249 A (INTEL CORP) 12 October 1995 (1995-10-12) * abstract; figure 1 * * claims 1-36 *	17-19	
Y	CA 2 187 855 A (COMPONENT ORIENTED PROTECTIVE) 13 June 1997 (1997-06-13) * abstract; figure 1 * * claims 1-20 * * page 9, line 23 - page 10, last line *	20	
Y	WO 95 24696 A (INTEGRATED TECH AMERICA ;MOONEY DAVID M (US); WOOD DAVID E (US); K) 14 September 1995 (1995-09-14) * abstract; figure 3 * * page 2, line 26 - page 3, line 19 * * claims 1-20 *	21	
Y	EP 0 895 148 A (SIEMENS AG) 3 February 1999 (1999-02-03) * the whole document *	22	

TECHNICAL FIELDS SEARCHED (Int.Cl.7)
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	17 March 2000	Powell, D

CATEGORY OF CITED DOCUMENTS
X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 30 4165

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

17-03-2000

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9845778 A	15-10-1998	AU 6850798 A	30-10-1998
US 5404532 A	04-04-1995	NONE	
WO 9527249 A	12-10-1995	AU 2237995 A	23-10-1995
		CN 1149343 A	07-05-1997
		EP 0754321 A	22-01-1997
		JP 10501907 T	17-02-1998
CA 2187855 A	13-06-1997	NONE	
WO 9524696 A	14-09-1995	US 5610981 A	11-03-1997
		AT 175505 T	15-01-1999
		AU 703856 B	01-04-1999
		AU 2092695 A	25-09-1995
		BR 9506968 A	01-06-1999
		CA 2183759 A	14-09-1995
		CN 1146813 A	02-04-1997
		DE 69507129 D	18-02-1999
		DE 69507129 T	05-08-1999
		EP 0748474 A	18-12-1996
		NZ 282954 A	24-11-1997
EP 0895148 A	03-02-1999	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82